1623

COMPUTER COMPANION FOR THE SINCLAIR/TIMEX COMPUTERS

BY ROBERT P. HAVILAND

COMPUTER COMPANION FOR THE SINCLAIR/TIMEX COMPUTERS

BY ROBERT P. HAVILAND

TAB TAB BOOKS Inc.
BLUE RIDGE SUMMIT, PA. 17214

FIRST EDITION

FIRST PRINTING

Copyright © 1983 by TAB BOOKS Inc.

Printed in the United States of America

Reproduction or publication of the content in any manner, without express permission of the publisher, is prohibited. No liability is assumed with respect to the use of the information herein.

Library of Congress Cataloging in Publication Data

Haviland, Robert P.
Computer companion for the Sinclair/Timex computers.

Includes index.
1. Timex 1000 (Computer) 2. Timex Sinclair 2000 (Computer) I. Title.
QA76.8.T48H38 1983 001.64 83-4965
ISBN 0-8306-1623-3 (pbk.)

Contents

Introduction		v
Reserved Words, arranged alphabetically		1
Appendix A	ZX81 Priority Table	79
Appendix B	ZX81 Character Codes	81
Appendix C	Graphic Symbols	85
Appendix D	System Variable Locations	87
Appendix E	ZX81 Storage Plan	89
Appendix F	Digit Assignment in Flags	91
Appendix G	Memory Map, ZX81 Family	93
Appendix H	Hex-Decimal Converter	95

Appendix I	Relative Jump Count Converter	97
Appendix J	Z80 Op Codes	99
Appendix K	Z80 Codes	103
Appendix L	References	117

Introduction

This little manual is intended to be a constant companion to the ZX80 family of personal computers, for use in creating programs and in getting them to run properly.

Each index tab opens to a keyword, all those used by the ZX81 and the others that have an 8K ROM. For each, there is an entry for

—the key and mode used for entry
—the storage code
—the required use form
—the conditions and results of use
—the report code likely to be encountered

Additionally, some entries give examples, hints and cautions.

The following terminology has been used.

Function —A function is evaluated at run time

and a single value returned.

Example: SIN X

Command—A command causes the computer, or a computer run, to accomplish a specific operation.

Example: STOP

Instruction—An instruction provides information to the computer, usually relating to program execution.

Example: TO

Expression—An expression is a combination of variables, constants, functions and operators, and possibly commands, which are reduced to a single value at run time.

Example: IF Y = SIN X THEN STOP

Literal—The result of a mode of the computer in which input is accepted without regard to class (numerical, etc. However the program itself may place conditions on items accepted).

Mode—Mode indicates the type of input the computer expects, indicating this by an inverse character cursor. Modes are K (Key word); L (Literal); F (Function); G (Graphics). L in quotes indicates String, and the report code (%, etc.), key word.

Run Time—The instant at which a specified item is being "worked on" by the computer's central processor.

Example: Normally, Line 20 run time follows Line 10 run time.

Functions and expressions are understood to be numerical unless stated. Other classes are logical, string, and graphics.

The ZX81 makes no distinction between integer numbers and those with decimals (floating point) for input or output. It will automatically choose the exponential form for very large or very small numbers. (Integer arithmetic is performed internally for some operations, such as those dealing with subscripts. Error signals will result in all cases if integers are outside the range 0-255 or 0-65535, as set by the operation.)

The ZX81 is unique in its use of single keystroke entries of key words, actually by immediate conversion of the keystroke to an internal number code. As a result, there are no restrictions on variable names—STOP, spelled with four letters, is a valid variable name. However, only a letter, or a letter followed by letters or numbers are allowed in names.

The ZX81 is also unique in checking program lines for logical (syntax) errors before transferring these from the screen bottom "work space" to the screen top "list" space. This check will not catch errors that involve two or more program lines.

In addition to the key words there are also appendices giving the character code used in Sinclair BASIC and the machine code and assembly code for the Z80 processor chip used by the Sinclair family. This appendix uses the assembly Op Codes as the indexing factor. A convenient

Hex-Decimal conversion table is also given.

In preparing this material, the ultimate authority has been the ZX81 Computer in 1K and 16K form, with checks run using 64K. The manufacturer's literature is the next level of authority, and general literature is the last. All stated forms and conditions have been checked and are correct at the time of writing.

The author, editor, and publisher would appreciate a note if you find any errors, or indications of change by the manufacturer, or have suggestions for making the volume more useful.

Code 210
(D2)h

Name Absolute value

Class Function

Key F on G

ABS

Form ABS value

Conditions Returns the magnitude of the value.

Value may be a number, a variable, or an expression.

ABS has no meaning for string expressions.

Examples ABS $X = |X|$
ABS -3.2 = ABS $3.2 = 3.2$

Report S - Not applicable to strings.

Code 203 (CB)h **Name** Arc cosine

Class Function **Key** [F] on S

ACS

Form ACS value

Conditions Returns the arc cosine of a value, i.e., an angle in radians.

The value may be a number, a variable, or an expression.

The value must be in the range +1 to −1.

ACS has no meaning for string variables.

Example ACS 0.70710678 = 0.78539816
(i.e., arc cos cos 45° = $\pi/4$ radians)

Reports [S] - Not applicable to String Values.
A - Invalid Argument.

Code 218
(DA)h

Class Logic Function

Key [K] or [L]
on shift 2

AND

Form Relation (1) AND relation(2)

Conditions Is a logical operator.

The right relation must be numeric.

The relations may be literals, a variable, or an expression.

There may be multiple logic functions in a single logical expression.

Examples AND NOT = nand operator

X AND Y gives
- X if Y is non-zero (True)
- 0 if Y is zero (False)

X$ AND Y gives
- X$ if Y is non-zero (True)
- " " if Y is zero (False)

where " " indicates the empty string.

Report Code [S]- Missing variable.

Code 202
(CA)h

Name Arc sine

Class Function

Key [F] on A

ASN

Form ASN value

Conditions Returns the arc sine of a value, i.e., an angle in radians.

The value may be a number, a variable, or an expression.

The value must be in the range +1 to −1.

ASN has no meaning for string elements.

Example ASN 0.70710678 = 0.78539816
i.e. ARC SIN 45° = $\pi/4$

Report Codes [S] - Not applicable to strings.

A - Invalid argument.

Code 193
(C1)h

Class Print Instruction **Key** [F] on C

AT

Form PRINT AT line, column; print-list.

Conditions Moves the location of the next print item to the line and column stated.

Lines are numbered 0-21 from the top of the screen to the bottom, columns are numbered 0-31 from the left of the screen to right.

For an explanation of the print-list see PRINT.

Will overwrite any characters already at the specified positions.

Example PRINT AT 10,15; "**"; (places two stars at screen center)

Report Code B - Integer out of range.

Code 204
(CC)h

Name Arc tangent

Class Function

Key [F] on D

ATN

Form ATN value

Conditions Returns the arc tangent of a value, i.e, an angle in radians.

The value may be a number, a variable or an expression.

ATN has no meaning for string expressions.

Example ATN 1 = 0.78539816
i.e., ATN TAN ($\pi/4$) = $\pi/4$

Report Code [S] - Not applicable to strings.

Code 0
(00)h

Class Command

Key Space
(during execution)

BREAK

Conditions Interrupts program execution if space key is pressed.

—at end of each BASIC line
—during LOAD, SAVE, COPY, PRINT or LPRINT.

There may be a delay until key depression is recognized.

Operation is resumed by CONT; however, the screen is cleared and the print position is reset to 0, 0.

Report Code D/line number

Code 214
(D6)h

Name Characters

Class Function

Key [F] on U

CHR$

Form CHR$ value

Conditions When applied to a numerical value, CHR$ returns the character whose code is that value, rounded to the nearest integer.

The value may be a number, a variable, or an expression, and must be in the range 0-255.

CHR$ has no meaning for strings.

Examples
CHR$ 40.2 = C
CHR$ 40.9 = D
CHR$ 200 = COS

Report Codes
[S] Not valid for strings.
B Value out of range.

Code 253
(FD)h

Class Command **Key** [F] on X

CLEAR

Conditions Used to delete all variables and to release the memory space they occupy.

Does not affect any program in storage.

Hint In programs that fill memory, use CLEAR when possible; otherwise delete variables no longer needed to release memory.

Code 251
(FB)h

Name Clear Screen

Class Command

Key [K] on V

CLS

Conditions Clears the screen.

Clears the display file; if there is less than 3.25K of memory, CLS sets the display file to just 25 ENTERs; if there is more than 3.25K of memory, the display file is set to an ENTER plus 24 lines each of 32 spaces plus an ENTER.

All commands except COPY effectively have CLS as first operation.

Hints Short programs, especially ones with SCROLL, may execute faster with RAMTOP set to 3K.

During program entry, clear the work space by using EDIT followed by the ENTER key.

Code 196
(C4)h

Class Function **Key** [F] on I

CODE

Form CODE "string quantity"

Conditions Returns the code of the first character of a string quantity which may be a literal, a string, or a string variable; 0 is returned if the string is empty.

The code of the nth character may be returned by the form Code A(N).

CODE has no meaning for numericals.

Examples CODE "ABC" = CODE "A" = 38 IF X$ = "ABC"

CODE X$(2) = CODE "B" = 39

Report Code [S] - Not applicable to numbers.

Code 232
(EB)h

Name Continue

Class Command

Key [K] on C

CONT

Conditions Causes resumption of program execution after any report a/b, where a is not zero.

Its effect is:

—GOTO b if a < > 9
—GOTO(b+1) if a = 9

Clears the display file.

Exception CONT is not effective if Report Code 5/n occurs during LIST. The form LIST n+1 may be used.

Code 255
(FF)h

Class Command **Key** [K] on Z

COPY

Conditions If Sinclair printer is connected, prints out a copy of the screen display.

If no printer, no action occurs except a brief flash of fast mode.

Caution A space prior to COPY causes the screen to be cleared.

Report Code D if BREAK is given.

Code 200
(C8)h

Name Cosine

Class Function

Key [F] on W

COS

Form COS angle

Conditions Returns the cosine of an angle which is radians.

The angle may be given by a number, a variable or an expression.

COS has no meaning for string quantities.

Examples COS $\pi = -1$
COS (π*2) = COS (2*π) = 1

Report Code [S] - Not applicable to strings.

Code 119
(77)h

Class Command **Key** on shifted 0

DELETE

Conditions Erases Character just prior to the work space cursor, G K or L

Cursor positioning is not possible in graphics mode, so return to L if any other deletion is required.

Notes To delete a workspace line, Use EDIT, ENTER in sequence.

To delete an entered program line, type its line number and press the enter key.

Code 233
(E9)h

Name Dimension

Class Command

Key [K] on D

DIM

Forms DIM single letter variable name $(N_1,...N_n)$
DIM single letter variable name $\$(N_1,...N_n)$

Conditions Establishes an array of dimensions specified by the Ns and reserves memory space equal to the product of the Ns.

DIM statements may occur at any place in a program, but they must occur before any operation with the array is attempted. Early in the program is the safest place.

The last subscript is varied first, for example

DIM B(2,3) gives the elements
B(1,1), B(1,2), B(1,3)
B(2,1), B(2,2), B(2,3)

A different value can be assigned to each element.

Each element of a numerical array

holds 4 bytes of mantissa plus sign, and 1 byte of exponent plus sign.

Each element of string array holds one character; in assignment, strings are truncated to fit dimensions, or are filled out with spaces.

When elements of string arrays are used, the last subscript of a string array may be omitted, giving a string of N_n characters.

When elements of string arrays are used the second subscript may be written as (,aTOb), giving a sliced string starting at a and ending at b, for example

If A$ (2,6) = RSTUVW, then A$ (2,3TO5) gives TUV

Either a or b may be omitted, causing the sliced string to start or end at the initial or final string position.

A simple numerical variable and a numerical array, can have the same letter name, but two arrays cannot, even though the dimensions are different. A string variable cannot have the same name as a string array.

The Ns may be numbers, variables or expressions, but they must be evaluatable at assignment time.

Report Codes 4 - No room in memory.
B - Integer out of range.

Code 117
(75)h

Class Command **Key** on shift 1

EDIT

Conditions Copies current line into workspace (the line pointed to by the cursor ▷ or the next line if pointed line is vacant).

The line may now be modified and reentered.

Use the sequence EDIT followed immediately by the enter key to clear the workspace.

Code 118
(76)h

Class Command **Key** ENTER

ENTER

Conditions If the item in the workspace meets the syntax test, ENTER initiates execution if there is no line number, or places the item in the program area of memory if it begins with a line number.

If [F] or [G] are in the workspace, ENTER resets the mode to the normal one.

After a report code, ENTER initiates list from the STOP line.

Note This key is labeled NEWLINE on the ZX80 and MicroAce®.

Report Code [S] indicates a syntax error just prior to its location.

Code 206 (CE)h

Name Exponent

Class Function

Key [F] on X

EXP

Form EXP number.

Conditions Raises the base of natural logarithms, e = 2.7182818, to the power of the number.

The number may be a number, a variable, or an expression.

For e, use the form, EXP 1.

Example EXP 2.7 = 14.879732

Note Maximum value of the number is approximately 88.

Report Code 6 - Arithmetic overflow.

Code 229
(E5)h

Class Command

Key on Shift F

FAST

Conditions Places computer in compute only mode.

Increases execution speed by 6-7 times by test.

Eliminates TV sync and picture.

Is used as a subroutine in COPY, LLIST, and SAVE.

Terminated by SLOW

A temporary return to SLOW is made during INPUT, PAUSE and with any report code.

Hint Use FAST during program and data entry. It is faster, and there is a screen flash at key action.

Code 235
(EB)h

Class Instruction **Key** [K] on F

FOR

Forms FOR variable = value 1 TO value 2
FOR variable = value 1 TO value 2 STEP value 3.

Conditions The initial statement in a FOR-NEXT loop.

The (control) variable must have a single letter name.

Value 1 is initially assigned to the control variable. Processing continues until "NEXT variable" is encountered. The variable is incremented by STEP value, or by 1 if not given. If the new value is less than value 2, the loop is repeated; if it is greater, processing jumps to the statement after "NEXT variable."

FORs and NEXTs must be equal in number.

One FOR-NEXT loop must be completely inside or completely outside another.

Values may be numbers, variables, or expressions, and need not be integers.

Control variables may be reused if loops are not nested.

Jumps to within a loop are likely to cause program errors or termination.

Hint It is often convenient to reserve variables I, J, K for array processing, and M, N for counter use.

Report Code 4 - Not enough room in memory.

Code 121
(79)h

Class Command

Key [K] [L] on Shift ENTER

FUNCTION

Conditions One of the four states of the keyboard, represented by [F] as the screen cursor.

Used to enter the functions listed on the bottom of the keys.

State is terminated by any key depression, but unmarked keys are ignored.

Code 237 (ED)h **Name** Go to subroutine

Class Instruction **Key** [K] on H

GOSUB

Form GOSUB line number

Conditions Causes a jump to the line indicated.

The current line is recorded, and program execution returned to it when RETURN is encountered.

If the line number does not exist, control is transferred to next line after that.

The line number may be a number, a variable, or an expression.

Subroutines may call other subroutines.

Hint Programs run faster if subroutines are placed at the start of the program.

Report Codes

0 - Called number greater than any program line.

4 - More GOSUBs than RETURNs.

B - Called number greater than 9999 (not a syntax error).

Code 236
(EC)h

Class Instruction **Key** [K]on G

GOTO

Form GOTO line number

Conditions Causes a transfer of program execution from the current line to target line, or if this is not present, to the next numbered line in the sequence.

The search for the target starts at line 0.

The line number may be a number, a variable, or an expression.

Hints Use GOTO 1 instead of RUN if initial values of variables are needed.

Use expressions to develop computed GOTOs.

Report Code 0 - Target line number greater than any in program.

Code 116
(74)h

Class Command

Key K / L on shift 9

GRAPHICS

Conditions One of the four states of the keyboard, indicated by the G cursor.

Used to enter inverse video or the 22 graphic symbols.

When entered, the state remains until shift 9 or the enter key is pressed.

Note An inverse character in the name during SAVE indicates "end of name".

Code 250
(FA)h

Class Instruction **Key** [K] on U

IF

Form IF condition THEN action

Conditions Causes execution of the action if the condition is True.

The condition may use the six relational operators, =, <, >, < =, > =, < >, or the logical operators AND, OR, NOT, with numerical or string values allowed.

The action must be an executable statement, with a keyword in its required form.

Note Compound conditions using AND, OR are allowed, but they must result in only a single action.

The shortened forms GOTO line number or THEN line number are not allowed. The statement must be THEN GOTO line number.

Code 65
(41)h

Class String Function **Key** [F] on B

INKEY$

Conditions Reads the keyboard and gives the character of the [L] mode or an empty string if no key is pressed.

Does not wait for a keystroke, but may be used in loops if multiple inputs are needed.

Space or L can be used to break execution.

Code 238
(EE)h

Class Command **Key** [K] on I

INPUT

Form INPUT variable name

Conditions When encountered in a program, places [L] (literal) or "[L]" (string literal) on the screen, and waits for input of data and ENTER.

Defines simple variables, but arrays must be separately dimensioned.

Cannot be used in direct execution mode.

To halt execution, use STOP in [L], or DELETE, STOP in "[L]" modes.

Report Codes

[S] - Wrong class of input.

8 - Cannot use in direct mode.

D - First character input was STOP.

Code 207
(CF)h

Name Integer

Class Function

Key [F] on R

INT

Form INT numerical value

Conditions Returns the next smaller integer of the value (rounds down).

The value may be a number, a variable, or an expression.

Has no meaning for string values.

Examples
```
INT PI = 3
INT – PI = –4
```

Hints Use form INT (value +0.5) to round to the nearest integer.

Use to "clean up" the screen.

Example - PRINT INT (100*PI)/100 prints 3.14 instead of 3.1415927.

Report Code [S] - not applicable to strings.

Code 198 (C6)h **Name** Length

Class String Function **Key** [F] on K

LEN

Form LEN "string"

Conditions Returns the length or number of characters in a string.

The string may be a literal, a string variable, or an expression (which must be in quotes).

The length of an empty string is 0; however, a space counts as a character, as do keyboard words.

LEN has no meaning for numerics.

Report Code [S]- Not applicable to numerics.

Code 241
(F1)h

Class Instruction **Key** [K] on L

LET

Form LET variable = value

Conditions Assigns the given value to the variable, but the value must be the same type as the variable.

Numerical values may be a number, a variable, or an expression.

String values may be a literal in quotes, a string variable, or an expression in quotes.

LET defines previously undefined variables.

LET cannot be omitted, and the form A = B = C is not allowed.

For dimensioned strings, the value is truncated to fit the dimension, or is filled out with blanks if short.

Code 240
(F0)h

Class Command **Key** [K] on K

LIST

Forms LIST

LIST line number

Conditions Produces a listing of an entered program.

The listing starts with the line number if it is given; otherwise it starts with line 0, or the next larger line if the line is nonexistent.

The line number may be a number, a variable, or an expression.

LIST n sets the [>] cursor to line n, even though it is an empty line. EDIT moves the line with the [>] cursor (or the next line) to the workspace.

ENTER after any report produces a listing.

During program entry, the system variable SCREENTOP is reset when the screen fills to keep the last line on the screen.

Hint The process of screen remaking can be shortened by commanding "LIST" last line entered" each ten lines or so.

Report Codes

4 - No room in memory.

5 - No room on screen.

Note CONT does *not* cause listing to resume after report code 5, but gives report code B - integer out of range. Use "LIST last screen line plus 1 instead.

Code 226
(E2)h

Name Line Print List

Class Command

Key [K] [L] on shift G

LLIST

Forms LLIST

LLIST line number

Conditions When the printer is connected, LLIST produces a printed program listing.

If no printer is connected, LLIST produces only a momentary FAST plus a clear screen.

The listing starts with line number if given; otherwise it starts with line 0, or the next larger line.

The line number may be a number, a variable, or an expression.

LLIST n sets the cursor [>] to line n, even though it is nonexistent.

Code 205 (CD)h **Name** Natural Logarithm

Class Function [F] on Z

LN

Form LN value

Conditions Returns the natural logarithm (to the base e = 2.7181828) of the value.

Value may be a number, a variable or an expression.

LN has no meaning for string variables.

Report Code A - Invalid argument.

Code 239
(EF)h

Class Command **Key** [K] on J

LOAD

Forms LOAD " "

LOAD "name"

Conditions Looks for a program called name on tape input (EAR plug), and places it in memory if found. The tape must be manually operated.

If name is the empty string, " ", the next program found is loaded.

See SAVE

BREAK interrupts loading and causes loss of current program if the new program has started loading.

Do not use inverse video in names, as all subsequent characters are lost.

Hints Program loading is more reliable when the name is used.

Tape input level is critical. Use a VU meter (approximately 2.0 volts), or an

LED flasher. TV bars should be clean and have high contrast.

For automatic program saving and re-execution, include the sequence

```
n       SAVE "name"
n + 1 GOTO start line
```

at the end of the program.

Report Code D - BREAK given.

Code 225 (E1)h **Name** Line Printer Print

Class Command **Key** [K][L] on S

LPRINT

Form LPRINT print-list

Conditions Used to output the contents of the print-list to a printer.

LPRINT produces only a brief screen flash if the printer is not connected.

See PRINT for print-list rules.

Lines are sent to the printer when:

- Print buffer is full (32 characters + RETURN).
- After a print-list not followed by comma or semicolon.
- When comma or TAB requires a new line.
- At the end of program, if the print-list is not complete.

Use BREAK to stop printing.

See also AT, TAB.

Report Code D - BREAK given.

Code 230
(E6)h

Class Command

Key [K] on A

NEW

Conditions Sets up the system for a new operation.

Clears program lines and variables from memory (location 16508 to RAMTOP).

At power on, RAMTOP is set to top of available memory, and is effectively followed by NEW.

Use CLEAR if the program is to be preserved but the variables cleared.

See RUN.

Code 243
(F3)h

Class Instruction **Key** [K] on N

NEXT

Form NEXT control variable

Conditions The second element of a FOR-NEXT loop.

The control variable must be a single letter name.

Finds the control variable, increments it by STEP size, and

—Jumps to the line after the line containing the FOR if the value of the variables is within limits, or
—Proceeds to line following NEXT.

See FOR, TO, STEP.

Hint The use of N as a control variable is especially convenient. It may be reused if the loops are independent.

Report Code 1 - No control variable.

Code 215
(D7)h

Class Logical Function **Key** [F] on N

NOT

Form NOT value

Conditions Is the negating logical operator, applicable to numerical values.

Value may be a number, a variable, or an expression.

Returns 1 if value is equal to zero (logically false) or 0 if value is equal to any other number.

Has no meaning for string values.

Code 217
(D9)h

Class Logical Function

Key [K] [L] shift on W

OR

Form Value 1 OR value 2

Conditions Is the logical alternate operator.

Values may be numbers, variables, or expressions.

OR has no meaning for strings.

There may be multiple ORs in a statement.

Example X OR Y has the value–
I if Y is not zero
X if Y is zero

Note The construct IF M = 1 OR M = 3 OR M = 5 THEN -- is legal.

Report Code [S] - Not applicable to strings.

Code 242
(F2)h

Class Command **Key** [K] on M

PAUSE

Form PAUSE interval

Conditions Creates a pause of the number of intervals specified.

Each interval is one TV frame or 1/50 second in duration.

Maximum duration is 32767 intervals. For a larger value, up to 65535, the pause is not timed, but lasts until a key is pressed.

BREAK during a pause terminates program execution.

Screen is displayed during a fast mode PAUSE.

For ZX80/MicroAce®, follow PAUSE by POKE 16437, 255.

Report Code D - BREAK given.

Code 211
(D3)h

Class Function **Key** [F] on O

PEEK

Form PEEK address

Conditions Returns the value of the byte at the specific address.

The address may be a number, a variable, or an expression.

The address must be in the range 0-65535 and is rounded to nearest integer for use.

Strings in the address must be preceded by VAL.

Report Code B - Values greater than 65535 or less than 0 not allowed.

Code 66
(42)h

Class Function (constant) **Key** [F] on M

PI

Conditions Returns the value 3.1415927.

Note The value $\pi/2$ is on a stack of constants, II is calculated as needed.

Hint Memory can be saved by using $\text{INT}\pi = 3$, $\text{NOT}\pi = 0$ and $\text{SGN}\,\pi = 1$.

Code 246
(F6)h

Class Graphics Command **Key** [K] on Q

PLOT

Form PLOT x-coordinate, y-coordinate

Conditions Blacks in the picture element at the specified coordinate and moves the print position to the following character position.

Coordinate origin is lower left of screen, positive numbers move the print position up and right. The maximums are x = 63, y = 43. The sign is ignored.

Coordinates may be a number, a variable, or an expression. The values are rounded to the nearest integer.

Note The point plotted by the PLOT command is as large as 1/4 of a character space. Points that occupy only 1/64 of a character space can be turned on or off using Machine Code.

Report Code B-Coordinate out of range.

Code 244
(F4)h

Class Command **Key** [K] on O

POKE

Form POKE address, value

Conditions Places value into the specified memory address.

Value must be "byte size", in the range −255 to +255. If negative, 256 is added before storage.

Address must be "two-byte size", in range 0-65535.

Value and address may be numbers, variables or expressions, and will be rounded to nearest integer.

Note Full understanding of POKE is necessary to full control of the computer and any add-ons.

Report Code B-Address, value out of range.

Code 245
(F5)h

Class Command **Key** [K] on P

PRINT

Forms PRINT

PRINT print-list

PRINT TAB column, print-list

PRINT AT line, column, print-list

Conditions Enters content of print-list into display file for screen display.

A print-list may be

- —Empty, producing a skipped line.
- —A number, which is printed.
- —A variable, whose current value is printed.
- —A string, which must be enclosed in quotes, displayed without the quotes; use " ", double apostrophes, to obtain quotes into a string.
- —Combinations of these items, separated by punctuation.
- —A semicolon causes continuous printing, with no spaces (a space in quotes is a valid character).
- —A comma causes at least one

space, plus enough additional space to start printing at column 0 or 16.

—Nothing (at the end of a line), causing the next PRINT to skip to the next line.

TAB causes the next item to be displayed on the current line if the current column is less than the one specified; otherwise a new line is started. The column may be specified by a number, a variable, or an expression, with values not exceeding 255, reduced module 32 for use.

AT causes printing to start at the specified line and column, erasing anything there. Line and column may be specified by a number, a variable, or an expression, line values can range from 0 through 21,and column values from 0 through 31.

Note It is worthwhile to experiment with print-list construction. For example, use PRINT list 1 , , list 2 instead of three separate PRINT lines.

See also AT, TAB, CLS, CONT, SCROLL

Report Codes

4 - Out of Memory.
5 - Screen Full.
B - Line, Column out of range.

Code 249 (F9)h **Name** Randomize

Class Command **Key** [K] on T

RAND

Form RAND seed

Conditions Controls the randomness of the pseudorandom function RND, by setting the seed, the system variable SEED.

Seed may be blank, a number, a variable, or an expression, and must be less than 65535.

If seed is zero or blank, the current value of the number of TV frames (system variable FRAMES) is used as the seed value.

See RND.

Report Code B - Seed out of range.

Code 234
(EA)h

Name Remark

Class Command

Key [F] on E

REM

Form REM any characters

Conditions REM is a command to place the line in the program listing, but to ignore the following characters (to the next ENTER) in execution

The primary uses of REM are to explain the program, identify the variables, and give the program name.

REM is useful as a place for machine code, which may be

- Hex characters.
- Sinclair codes, and regular and reverse characters.
- If not listed, any Sinclair code except HALT can be used.

Code 254
(FE)h

Class Command **Key** [K] on Y

RETURN

Conditions Is the necessary end statement of a subroutine.
Returns program execution to the line following GOSUB n.
The number of executed RETURNs and GOSUBs must be equal.

Report Codes
0 - may indicate the omission of a RETURN.
7 - GOSUB not found on return.

Code 64
(40)h

Name Random number

Class Function

Key [F] on T

RND

Conditions Returns one of a series of 65535 pseudorandom numbers, 0<=Rn<1.

The starting point of the sequence is set by RANDn.

Unless reset by RANDn, each successive RND returns the next number of the pseudorandom sequence.

Report Code [S] - RND cannot have an operand.

Code 247
(F7)h

Class Command **Key** [K] on R

RUN

Forms RUN

RUN line number

Conditions Initiates program execution from line specified or line 0 if not given.

Clears all variables and stacks before starting execution.

Use GOTO when variables must be preserved, as for initialization.

Hint Use RUNn to use independent sub-programs in a master program.

Report Code Syntax and other report codes may occur during a run.

Code 248
(F8)h

Class Command **Key** [K] on S

SAVE

Form SAVE "name"

Conditions Converts TV signal output to tape format for manual recording on magnetic tape.

Records name, plus system variables, screen display, program and program variables (to E-LINE).

Name is required, and can be any string, but characters after 1st inverse video character are lost.

The record is a six second pause followed by the bit pattern of 4 or 9 pulses, at a frame rate of 420 Baud.

Do not use save in GOSUBs.

Hints Record level is critical, and recorders with automatic level may give poor results. Make test records starting at a low level and increasing until maximum loudness is reached. On playback, the TV bars should be clear and black. Remove unused tape plugs during SAVE and LOAD.

To make programs self-saving use

```
n     SAVE "name"
n+1 GOTO start line
```

at the end of program, and initiate the SAVE routine with GOTO n.

BREAK interrupts SAVE.

See LOAD.

Report Codes

F - No name supplied.
D - BREAK given.

Code 231
(E7)h

Class Command **Key** [K] on B

SCROLL

Conditions When encountered, shift the display file up one line and move the print position to the start of the bottom line.

A line of more than 32 characters during scrolling interrupts program execution. CONT will display all of the line, but at the top of a clear screen.

Hints Combine with PAUSE to make moving displays.

Lateral scrolling is possible with machine language coding.

Report Code 5 - line too long (no room on screen).

Code 209
(D1)h

Name Signum

Class Function

Key F on F

SGN

Form SGN value

Conditions Returns the sign function (signum) of a value.

SGN has no meaning for string variables.

Examples SGN x = −1 if x is negative
SGN x = 0 if x is zero
SGN x = 1 if x is positive

Note The 8K ROM will not raise negative numbers to a power. Use SGN and ABS to determine sign and magnitude separately.

Code 199
(C7)h

Name Sine

Class Function

Key [F] on Q

SIN

Form SIN angle

Conditions Returns the sine of an angle expressed in radians.

The angle may be a number, a variable, or an expression.

SIN has no meaning for string variables.

Note Because of priorities in execution, SIN 2 $*\pi = 0$. Use parenthesis - SIN $(2 * \pi) =$ SIN $(\pi * 2) = 0$.

Report Code [S] - not applicable to strings.

Code 228
(E4)h

Class Command

Key [K] [L] on shift D

SLOW

Conditions Places computer in compute and display mode (ZX81 and TS 1000 only).

Since the CPU is occupied with the display during 24 picture lines, the time for computing is limited.

Execution speed is about 1/4 (according to Sinclair) or 1/6 - 1/7 (by test) that of the FAST mode.

See FAST.

Code 208
(D0)h

Name Square Root

Class Function

Key [F] on H

SQR

Form SQR value

Conditions Returns the square root of the value.

Value may be a number, a variable, or an expression.

Is not defined for negative values.

Has no meaning for string quantities.

Report Codes
[S] - Not applicable to strings.
A - Invalid argument.

Code 224
(E0)h

Class Instruction

Key [K] [L] on shift E

STEP

Form FOR variable value 1 TO value 2 STEP value 3

Conditions An optional part of a FOR-NEXT loop, used to set the increment to value 3.

If omitted, the increment is plus 1.

The increment may be nonintegral, and positive or negative, but cannot be changed during execution.

Value 3 may be a number, a variable, or an expression.

See FOR, TO, NEXT

Code 227
(E3)h

Class Command

Key on shift A

STOP

Conditions Halts program execution when encountered in a program or at the end of a program line, and during SAVE, COPY, LIST, LLIST if given from the keyboard.

Execution can be registered by CONT, but the screen is cleared.

Hints Use STOP to establish a breakpoint in programs; the variables can be examined by keyboard command.

Use with IF-THEN to establish error flags, such as negative SQR, divide by zero, etc.

Code 213 (D5)h **Name** String Function

Class String Function **Key** F on Y

STR$

Form STR$ value

Conditions Applied to a numerical value, returns the value as a string quantity.

Value may be a number, a variable, or an expression, the current or reduced value being returned.

The return may be manipulated by string rules.

Code 194 (C2)h **Name** Tabular set

Class (Print) Function **Key** [F] on P

TAB

Form TAB column;

Conditions An element of a print-list, used to set the print position to a specified column.

Column may be a number, a variable, or an expression. The maximum value is 255 and is reduced modulo 32 for use.

If backspace would be required, move the print position to the next line.

See PRINT.

Example PRINT TAB0; "*"; TAB 31; "*" places a star at each end of the current line, if the print position is 0, or the next line if it is not.

Report Codes [S]- missing punctuation.
B - Column out of range.

Code 201
(C9)h

Name Tangent

Class Function

Key [F] on F

TAN

Form TAN angle

Conditions Returns the tangent of an angle expressed in radians.

The angle may be a number, a variable, or an expression.

TAN has no meaning for string quantities.

Note Because of priorities in execution, TAN $2*\pi = 0$. Use parenthesis to obtain TAN $(2*\pi)$ = TAN $(\pi*2) = 0$.

Report Code [S]- not applicable to strings.

Code 222
(DE)h

Class Command

Key [K] [L] on shift 3

THEN

Form IF condition THEN keyword expression

Conditions The second part of the IF-THEN construct.

If the condition is true, the instruction following THEN is executed, otherwise processing proceeds to the next line.

The keyword cannot be omitted, but not all keywords require or allow a following expression.

See IF.

Code 223
(DF)h

Class Instruction

Key [K] [L] on shift 4

TO

Forms FOR control variable-initial value TO limit value STEP increment

Name $ (subscript 1 TO subscript 2)
"String" (subscript 1 TO subscript 2)
Name $ (subscript 1, subscript 2 TO subscript 3)

Conditions (a) Part of the FOR-NEXT loop construct.

See FOR, NEXT, STEP.

(b) The keyword which converts a string array subscript to a string slicer.

Applicable to strings or string arrays, but must relate to the last dimension (the form A$ (2 TO 5, n) is not allowed.

Causes the part prior to subscript 1 and after subscript 2 to be ignored.

Either or both subscripts can be omit-

ted; the missing part is then taken as the beginning or end of the string.

The two subscripts may be the same, giving a single string character.

The subscripts must not be negative.

If the first subscript is larger than the second, a null string results.

Examples

```
IF A$ = "QWERTY"
A$ (2 TO 4) = "WER"
A$ (TO 3) = "QWE"
A$ (3 TO 3) = A$(3) = E
A$ (4 TO 3) = "          "
```

Report Code

3 - Subscript out of range.

Code 252
(FC)h

Class Graphics Command **Key** [K] on W

UNPLOT

Form UNPLOT x-coordinate, y-coordinate

Conditions Deletes the picture element at the stated coordinate, if present.

Moves the print position to the following character position.

Coordinates may be numbers, variables or expressions.

Minus values are treated as IF positive.

Fractional values are rounded to the nearest integer.

Maximum values are X = 63, Y = 43.

The coordinate origin is the lower left of the screen.

See PLOT.

Report Code B - Coordinate out of range.

Code 212 (D4)h **Name** User routine

Class Function **Key** [F] on L

USR

Form USR address

Conditions Transfers program execution to the machine code routine at the address.

The routine must end with the Z80 code for RET, and the Ix and I registers must contain 4000h and 1Eh, respectively.

In FAST, the routine must not use a′, F′, IX, IY, and R registers, nor read the a′-f′ pair.

On RET, the bc pair may contain a two byte unsigned integer, which can be printed by a PRINT USR address call.

Note Machine code may be placed in a REM statement, above RAMTOP, or between 8K-16K and 32K-64K if expanded memory is used.

There are many callable routines in the ROM that are sometimes useful.

Warning Improper calls may place execution in an endless loop, (a crash), requiring power removal and loss of program, memory, and variables. Be certain that there are no such problems before using machine language in long programs. Also be sure that you have saved the program on tape before you attempt to run it.

Code 197 (C5)h **Name** Value function

Class String Function **Key** [F] on J

VAL

Form VAL string quantity

Conditions Applied to a string quantity, evaluates the quantity as a numerical.

The quantity may be a string in quotes, a string variable, or a string expression, and must be correct as a numerical.

If used in an expression, VAL must be the first item in the expression.

Can appear only in the first coordinate of PRINT AT, PLOT, and UNPLOT.

Example If A$ = "ATN 1*4"
VAL A$ gives 3.1415927

Report Code The string quantity does not form a valid numerical.

Appendix

Appendix A
ZX81 Priority Table

PRIORITY	FUNCTION OR OPERATION
2	OR
3	AND
4	NOT
5	<=,>=,<>,>,<,=
6	+, Binary –
8	*, /
9	Unary –
10	**
12	Subscripts, Slices
16	All other Functions

Notes The , above are separators only.

Highest priority items are executed first.

Use parenthesis to establish the desired order of execution.

A quantity in parenthesis must reduce to a single string or numerical value.

Appendix B
ZX81 Character Codes

First digits \ Last digit	0	1	2	3	4
00	SPACE	G	G	G	G
01	G	"	£	$	:
02	=	+	−	*	/
03	2	3	4	5	6
04	C	D	E	F	G
05	M	N	O	P	Q
06	W	X	Y	Z	RND
07	?	?	?	?	?
08	?	?	?	?	?
09	?	?	?	?	?
10	?	?	?	?	?
11	?	?	↑	↓	←
12	MODE	FUNCTION	?	?	?
13	G	G	G	G	G
14	£	$	:	?	(
15	−	*	/	;	)
16	4	5	6	7	8
17	E	F	G	H	I
18	O	P	Q	R	S
19	Y	Z	""	AT	TAB
20	COS	TAN	ASN	ACS	ATN
21	ABS	PEEK	USR	STR$	CHR$
22	>=	<>	THEN	TO	STEP
23	NEW	SCROLL	CONT	DIM	REM
24	LIST	LET	PAUSE	NEXT	POKE
25	IF	CLS	UNPLOT	CLEAR	RETURN

G-Graphics
□-Inverse Video

5	6	7	8	9
G	G	G	G	G
?	(	)	>	<
;	,	.	0	1
7	8	9	A	B
H	I	J	K	L
R	S	T	U	V
INKEY	PI	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
→	GRAPHICS	EDIT	ENTER	DELETE
?	?	?	G	G
G	G	G	G	"
)	>	<	=	+
.	0	1	2	3
9	A	B	C	D
J	K	L	M	N
T	U	V	W	X
?	CODE	VAL	LEN	SIN
LN	EXP	INT	SQR	SGN
NOT	**	OR	AND	<=
LPRINT	LLIST	STOP	SLOW	FAST
FOR	GOTO	GOSUB	INPUT	LOAD
PRINT	PLOT	RUN	SAVE	RAND
COPY				

Appendix C
Graphic Symbols

	Code	Key
	0	Space
	1	Shift 1
	2	Shift 2
	3	Shift 7
	4	Shift 4
	5	Shift 5
	6	Shift
	7	Shift E
	8	Shift A
	9	Shift D
	10	Shift S

Graphic Symbols continued.

	Code	Key
	128	Space
	129	Shift Q
	130	Shift W
	131	Shift G
	132	Shift R
	133	Shift 8
	134	Shift Y
	135	Shift 3
	136	Shift H
	137	Shift G
	138	Shift F

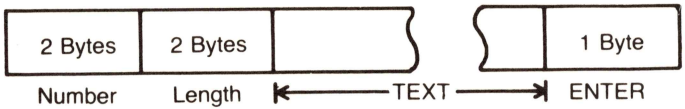
Note: All are available in [G] mode except
[K] or [L] for Code 0 (white)

Appendix D
System Variable Locations

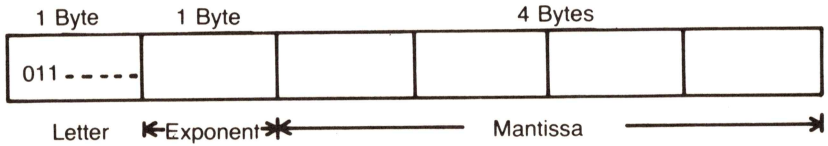
16384	ERR-NR	16419	S-TOP
16385	FLAGS	16421	LAST-K
16386	ERR-SP	16423	DEBOUNCE
16388	RAMTOP	16424	MARGIN
16390	MODE	16425	NXTLIN
16391	PPC	16427	OLDPPC
16393	VERSN	16429	FLAGX
16394	E-PPC	16430	STRLEN
16396	D-FILE	16432	T-ADDR
16398	DF-CC	16434	SEED
16400	VARS	16436	FRAMES
16402	DEST	16438	COORDS
16404	E-LINE	16440	PR-CC
16406	CH-ADD	16441	S-POSN
16408	X-PTR	16443	CDFLAG
16410	STKBOT	16444	PRBUFF
16412	STKEND	16477	MEMBOT
16414	BREG		
16415	MEM	16417	
16418	DF—SZ	16507	not used
		16508	

Appendix E
ZX81 Storage Plan

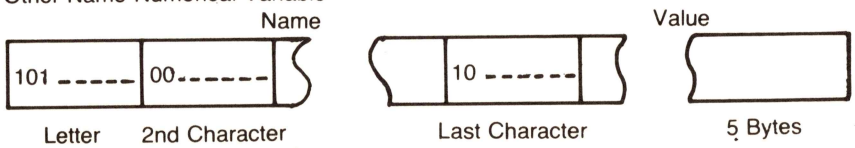
Program Lines



Single Letter Numerical Variable

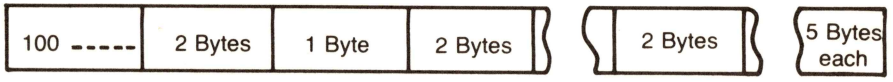


Other Name Numerical Variable



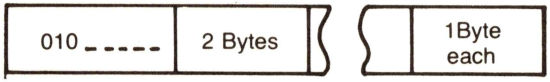
ZX81 Storage Plan continued.

Numerical Array: For string arrays, prefix starts with 010, and there is 1 byte per element



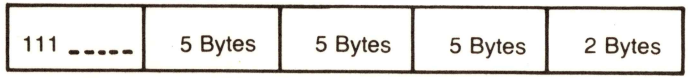
Letter −20n | elements × dimensions +1 | No. of dimensions | 1st dimensions | last dimension | elements

String



Letter −20h | Number characters | ← Characters →

FOR-NEXT Control Variable



Letter | Value | Limit | Step | Looping line

Note!
Looping line is stored high byte first.

Appendix F
Digit Assignment in Flags

Flag S - 16385

Bit 0	S - no space before token	R - space
Bit 1	S - pass to printer	R - to screen
Bit 2	S - K or L mode	R - F or G
Bit 6	S - numbers	R - strings
Bit 7	S - check line syntax	R - no check

Flag X - 16429

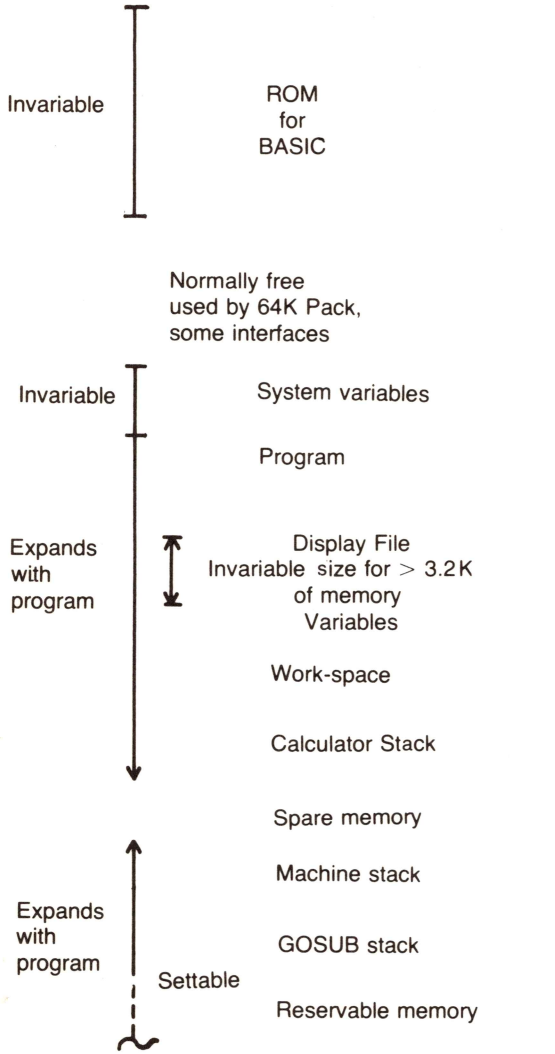
Bit 0	S - simple variable	R - array
Bit 1	S - variable free to use	R - used
Bit 5	S - input mode	R - edit
Bit 6	S - number	R - string

CD Flag - 16443

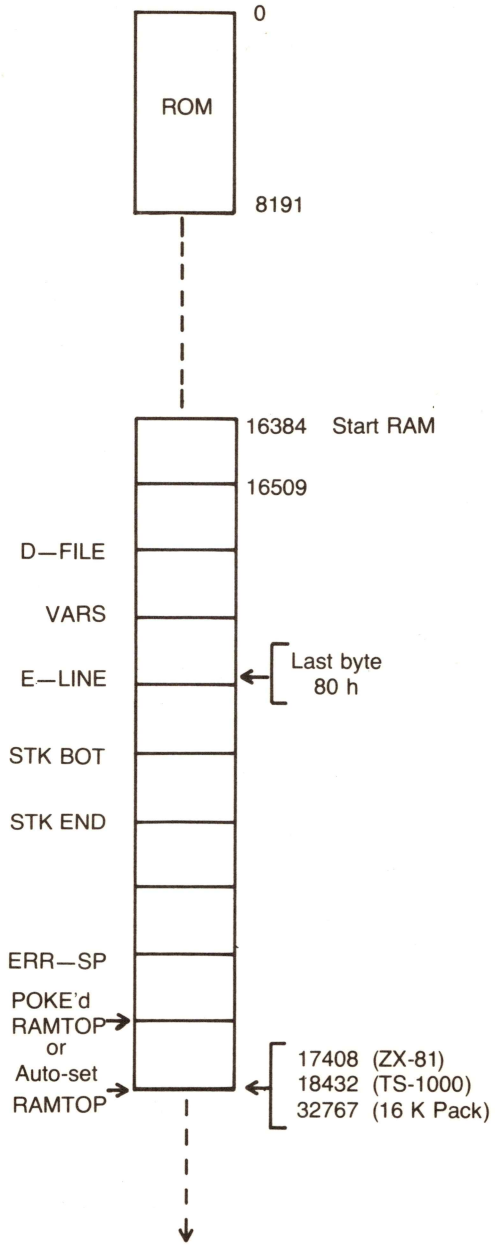
Bit 0	S - a key has been pressed	R - no key
Bit 6	S - slow mode	R - fast
Bit 7	S - slow mode copy	R - fast (temporary)

S - Bit is Set (1)
R - Bit is Reset (0)

Appendix G
Memory Map, ZX81 Family



Memory Map, ZX81 Family continued.



Appendix H
Hex-Decimal Converter

First Hex Digit \ Second Hex Digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Appendix I
Relative Jump Count Converter

Decimal - Hexadecimal

Note: Watch count beginning/end rules

Minus or
Back Jump

First Hex Digit

Second Hex Digit	F	E	D	C	B	A	9	8	Second Hex Digit
–	0	16	32	48	64	80	96	112	0
F	1	17	33	49	65	81	97	113	1
E	2	18	34	50	66	82	98	114	2
D	3	19	35	51	67	83	99	115	3
C	4	20	36	52	68	84	100	116	4
B	5	21	37	53	69	85	101	117	5
A	6	22	38	54	70	86	102	118	6
9	7	23	39	55	71	87	103	119	7
8	8	24	40	56	72	88	104	120	8
7	9	25	41	57	73	89	105	121	9
6	10	26	42	58	74	90	106	122	A
5	11	27	43	59	75	91	107	123	B
4	12	28	44	60	76	92	108	124	C
3	13	29	45	61	77	93	109	125	D
2	14	30	46	62	78	94	110	126	E
1	15	31	47	63	79	95	111	127	F
0	16	32	48	64	80	96	112	128	+
	0	1	2	3	4	5	6	7	

First Hex Digit

Plus or
Forward Jump

Appendix J
Z80 Op Codes

Note: These are extensions to Appendix A in the Sinclair ZX81 Manual
(Replace 1X by 1Y if following Code 253 (FD))

Follows code 221 (FD)

CODE	Z80 Op Code
9	ADD IX, BC
25	ADD IX, DE
33	LD IX, nn
34	LD (nn), IX
35	INC IX
41	ADD IX, IX
42	LD IX, (nn)
43	DEC IX
52	INC (IX+d)
53	DEC (IX+d)
54	LD(IX+d), n
57	ADD IX, SP
70	LD B, (IX+d)
78	LD C, (IX+d)
86	LD D, (IX+d)
94	LD E, (IX+d)
102	LD H, (IX+d)
110	LD L, (IX+d)
112	LD (IX+d), B
113	LD (IX+d), C
114	LD (IX+d), D
115	LD (IX+d), E
116	LD (IX+d), H
117	LD (IX+d), L
119	LD (IX+d), A
126	LD A, (IX+d)

Z80 Op Codes continued.

Code	Z80 Op Code
134	ADDA, (IX+d)
142	ADC A, (IX+d)
150	SUB A, (IX+d)
158	SBC A, (IX+d)
166	AND A, (IX+d)
174	XOR A, (IX+d)
182	OR A, (IX+d)
190	CP A, (IX+d)
203	See Below
225	POP IX
227	EX (SP), IX
229	PUSH IX
233	JP (IX)
249	LD SP, IX

Follows code 221, 203 DDCB

Code	Z80 Op Code
6	RLC (IX+d)
14	RRC (IX+d)
22	RL (IX+d)
30	RR (IX+d)
38	SLA (IX+d)
46	SRA (IX+d)
62	SRL (IX+d)
70	BIT 0, (IX+d)
78	BIT 1, (IX+d)
86	BIT 2, (IX+d)
94	BIT 3, (IX+d)
102	BIT4, (IX+d)
110	BIT 5, (IX+d)
118	BIT 6, (IX+d)
126	BIT 7, (IX+d)
134	RES 0, (IX+d)
142	RES 1, (IX+d)
150	RES 2, (IX+d)
158	RES 3, (IX+d)

Code	Z80 Op Code
166	RES 4, (IX+d)
174	RES 5, (IX+d)
182	RES 6, (IX+d)
190	RES 7, (IX+d)
198	SET 0, (IX+d)
206	SET 1, (IX+d)
214	SET 2, (IX+d)
222	SET 3, (IX+d)
230	SET 4, (IX+d)
238	SET 5, (IX+d)
246	SET 6, (IX+d)
254	SET 7, (IX+d)

Appendix K
Z80 Codes

( ) Register is pointer
+ See Bit Operations

RST/8	0	1	2	3	4	5	6	7									
CC-Condition	N2.	Z.	NC.	C.	PO.	PE.	P.	M.									
Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	n nn
ADC A,—	88	89	8A	8B	8C	8D	(8E)	8F							(DD8E) d	(FD8E) d	CE n
ADC HL,—									ED 4A	ED 5A	ED 6A	ED 7A					
ADD A,—	80	81	82	83	84	85	(86)	87							(DD86) d	(FD86) d	C6 n
ADD HL,—									09	19	29	39					
ADD IX,—									DD 09	DD 19		DD 39	DD 29				
ADD IY,—									FD 09	FD 19		FD 39	FD 29				
AND—	A0	A1	A2	A3	A4	A5	(A6)	A7							(DDA6) d	(FDA6) d	E6 n
BIT 6,—	CB +	CB +	CB +	CB +	CB +	CB +	(CB) +	CB +							(DDCB) d+	(FDCB) d+	
CALL cc,nn	C4 nn.	CC nn.	D4 nn.	DC nn.	E4 nn.	EC nn.	F4 nn.	FC nn.									

Z80 Codes continued.

( ) Register is pointer
\+ See Bit Operations

RST/8	0	1	2	3	4	5	6	7									
CC-Condition	N2.	Z.	NC.	C.	PO.	PE.	P.	M.									
Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	n nn
CALL nn																	CD nn
CCF																	3F
CP—	B8	B9	BA	BB	BC	BD	(BF)	BF							(DDBE) d	(FDBE) d	FE n
CPD																	ED A9
CP DR																	ED B9
CPI																	ED A1
CPIR																	ED B1

CPL																	2F
DAA																	27
DEC—	05	0D	15	1D	25	2D	(35)	3D	0B	1B	2B	3B	DD 2B	FD 2B	(DD35) d	(FD35) d	
DI																	F3
DJNZE																	10 E-2
EI																	FB
EX(SP),—											E3		DD E3	FD E3			
EX AF,AF'																	08
EX DE, HL																	EB
EXX																	D9
HALT																	76

Z80 Codes continued.

( ) Register is pointer

IM	0	1	2	3	4	5	6	7									
CC-Condition	NZ	Z	NC	C	PO	PE	P	M		I	R	Special Register					
Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	none n nn
IM—	ED 46	ED 56	ED 5E														
INA,(—)																	DB (n)
IN—,(C)	ED 40	ED 48	ED 50	ED 58	ED 60	ED 68		ED 78									
INC—	04	0C	14	1C	24	2C	(34)	3C	03	13	23	33	DD 23	FD 23	(DD34) d	(FD34) d	
IND																	ED AA
INDR																	ED BA
INI																	ED A2
INIR																	ED B2
JP ( )							(E9)						(DD- E9)	(FD- E9)			
JP cc,nn	C2 nn	CA nn	D2 nn	DA nn	E2 nn	EA nn	F2 nn	FA nn									
JP—																	C3 nn
JR cc, e	20 e-2	28 e-2	30 e-2	38 e-2													18 e-2
LDB,—	40	41	42	43	44	45	(46)	47							(DD46) d	(FD46) d	06 n

LDC,—	48	49	4A	4B	4C	4D	(4E)	4F							(DD4E) d	(FD4E) d	0E n
LDD,—	50	51	52	53	54	55	(56)	57							(DD56) d	(FD56) d	16 n
LDE,—	58	59	5A	5B	5C		(5E)	5F							(DD5E) d	(FD5E) d	1E n
LDH,—	60	61	62	63	64	65	(66)	67							(DD66) d	(FD66) d	26 n
LDL,—	68	69	6A	6B	6C	6D	(6E)	6F							(DD6E) d	(FD6E) d	2E n
LDA,—	78	79	7A	7B	7C	7D	(7E)	7F	(0A)	(1A)					(DD7E) d	(FD7E) d	3E n
LDA,—										ED 57	ED 5F						3A (nn)
LD(BC),—								(02)									
LD(DE),—								(12)									
LD(HL),—	(70)	(71)	(72)	(73)	(74)	(75)		(77)									(36) n
LD(IY+d),—	(FD-70) d	(FD-71) d	(FD-72) d	(FD-73) d	(FD-74) d	(FD-75) d		(FD-77) d									(FD36) d,n
LD(IX+d),—	(DD-70) d	(DD-71) d	(DD-72) d	(DD-73) d	(DD-74) d	(DD-75) d		(DD-77) d									(DD 36) d,n
LDI, A								ED 47									

Z80 Codes continued.

( ) Register is pointer
+ See Bit Operations

	0	1	2	3	4	5	6	7				AF	Register			(nn)	
CC-Condition	NZ	Z	NC	C	PO	PE	P	M									
Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	none n nn
LDR,A								ED 4F									
LD BC,—																ED4B (nn)	01 nn
LDDE,—																ED58 (nn)	11 nn
LDHL,—																2A (nn)	21 nn
LD SP,—											F9		DD F9	FD F9		ED7B (nn)	31 nn
LD IX,—																	DD21 nn
LD IY,—																	FD21 nn
LD(nn),—								32 (nn)	ED43 (nn)	ED53 (nn)	22 (nn)	ED73 (nn)	DD22 (nn)	FD22 (nn)			
LDIX,(—)																DD2A (nn)	D021 nn
LDIY,(—)																FD2A (nn)	FD21 nn
LDD																	EDA8
LDDR																	EDB8

LDI																	EDA0
LDIR																	EDB0
NEG																	ED44
NOP																	00
OR—	B0	B1	B2	B3	B4	B5	(B6)	B7							(DDB6) d	(FDB6) d	F6 n
OTDR																	ED BB
OTIR																	ED B3
OUT(C),—	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69		ED 79									
OUT(n),A								D3 (n)									
OUT D																	ED AB
OUT I																	ED A3
POP—									C1	D1	E1	F1	DD E1	FD E1			
PUSH—									C5	D5	E5	F5	DD E5	FD E5			
RESb,—	CB +	CB +	CB +	CB +	CB +	CB +	(CB) +	CB +							(DDCB) d+	(FDCB) d+	
RET																	

Z80 Codes continued.

( ) Register is pointer
+ See Bit Operations

RST18	0	1	2	3	4	5	6	7									
CC-Condition	NZ	Z	NC	C	PO	PE	P	M									
Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	none n nn
RET—	C0	C7	D0	D7	E0	E7	F0	F7									
RETI																	ED 4D
RETN																	ED 5D
RL—	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	(CB) 16	CB 17							(DDCB) d 16	(FDCB) d 16	
RLA																	17
RLC—	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	(CB) 06	CB 07							(DDCB) d 06	(FDCB) d 06	
RLCA																	07
RLD																	ED 6F
RR—	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	(CB) 1E	CB 1F							(DDCB) d 1E	(FDCB) d 1E	
RRA																	1F

RRC—	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	(CB) 0E	CB 0F							(DDCB) d 0E	(FDCB) d 0E	
RRCA																	0F
RRD																	ED 67
RST—	C7	CF	D7	DF	E7	EF	F7	FF									
SBCA,—	98	99	9A	9B	9C	9D	(9E)	9F							(DD9E) d	(FD9E) d	DE n
SBC HL,—									ED 42	ED 52	ED 62	ED 72					
SCF																	37
SET b,—	CB +	CB +	CB +	CB +	CB +	CB +	(CB) +	CB +							(DDCB) d +	(FDCB) d +	
SLA—	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	(CB) 26	CB 27							(DDCB) d 26	(FDCB) d 26	
SRA—	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	(CB) 2E	CB 2F							(DDCB) d 2E	(FDCB) d 2E	
SRL—	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	(CB) 3E	CB 3F							(DDCB) d 3E	(FDCB) d 3E	
SUB—	90	91	92	93	94	95	(96)	97							(DD96) d	(FD96) d	D6 n

Z80 Codes continued.

RST18	0	1	2	3	4	5	6	7
CC-Condition	NZ	Z	NC	C	PO	PE	P	M

( ) Register is pointer
+ See Bit Operations

Register	B	C	D	E	H	L	(HL)	A	BC	DE	HL	SP	IX	IY	(IX+d)	(IY+d)	none n nn
XOR—	A8	A9	AA	AB	AC	AD	(AE)	AF							(DDAE) d	(FDAE) d	EE n

Bit Operations

Register	B	C	D	E	H	L	(HL)	A								(IX+d)	(IY+d)
BIT 0,—	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	(CB) 46	CB 47								(DDCB) d 46	(FDCB) d 46
BIT 1,—	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	(CB) 4E	CB 4F								(DDCB) d 4E	(FDCB) d 4E
BIT 2,—	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	(CB) 56	CB 57								(DDCB) d 56	(FDCB) d 56
BIT 3,—	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	(CB) 5E	CB 5F								(DDCB) d 5E	(FDCB) d 5E
BIT 4,—	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	(CB) 66	CB 67								(DDCB) d 66	(FDCB) d 66
BIT 5,—	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	(CB) 6E	CB 6F								(DDCB) d 6E	(FDCB) d 6E
BIT 6,—	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	(CB) 76	CB 77								(DDCB) d 76	(FDCB) d 76
BIT 7,—	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	(CB) 7E	CB 7F								(DDCB) d 7E	(FDCB) d 7E
RES 0,—	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	(CB) 86	CB 87								(DDCB) d 86	(FDCB) d 86
RES 1,—	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	(CB) 8E	CB 8F								(DDCB) d 8E	(FDCB) d 8E

Z80 Codes continued.

Bit Operations

Register	B	C	D	E	H	L	(HL)	A							(IX+d)	(IY+d)
RES 2,—	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	(CB) 96	CB 97							(DDCB) d 96	(FDCB) d 96
RES 3,—	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	(CB) 9E	CB 9F							(DDCB) d 9E	(FDCB) d 9E
RES 4,—	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	(CB) A6	CB A7							(DDCB) d A6	(FDCB) d A6
RES 5,—	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	(CB) AE	CB AF							(DDCB) d AE	(FDCB) d AE
RES 6,—	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	(CB) B6	CB B7							(DDCB) d B6	(FDCB) d B6
RES 7,—	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	(CB) BE	CB BF							(DDCB) d BE	(FDCB) d BE
SET 0,—	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	(CB) C6	CB C7							(DDCB) d C6	(FDCB) d C6
SET 1,—	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	(CB) CE	CB CF							(DDCB) d CE	(FDCB) d CE
SET 2,—	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	(CB) D6	CB D7							(DDCB) d D6	(FDCB) d D6
SET 3,—	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	(CB) DE	CB DF							(DDCB) d DE	(FDCB) d DE

SET 4,—	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	(CB) E6	CB E7								(DDCB) d E6	(FDCB) d E6
SET 5,—	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	(CB) EE	CB EF								(DDCB) d EE	(FDCB) d EE
SET 6,—	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	(CB) F6	CB F7								(DDCB) d F6	(FDCB) d F6
SET 7,—	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	(CB) FE	CB FF								(DDCB) d FE	(FDCB) d FE

Appendix L
References

—ZX81 BASIC PROGRAMMING,
Steve Vickers, Sinclair, 1981
—Sinclair ZX81 ROM Disassembly,
Ports A and B. Ian Logan, Melbourne House, 1981
—The ZX81 Companion, Robert Mannder,
Creative Computing, 1981
—Mastering Machine Code on your ZX81,
Toni Baker, Reston, 1982
—BASIC Programming, John G. Kemeny
and Thomas E. Kurtz, Wiley, current ed.
—Z80 Assembly Language Programming
Manual, Zilog, current ed.

Computer Companion for the Sinclair/Timex Computers

by Robert P. Haviland

- Instant access to all the information you need to create programs and get them running smoothly!
- All the key words used by the Timex-Sinclair 1000, Sinclair ZX81 and Z80, and Microace, listed in easy-to-use alphabetical order!
- Includes key and mode used for entry, the storage code, required use form, conditions and results of use, and report codes plus use examples, tips, and cautions!
- Each key word is listed on a separate page . . . with tab index for quick reference!
- Lie-flat comb binding for easy use . . . printed in large, easy-to-read type on heavy card stock!

Now you can have all the information you need to program your Timex/Sinclair machine right at your fingertips . . . when and where you need it! This exceptional sourcebook is designed for quick and easy reference . . . the perfect way to refresh your memory on available terms and how those terms should be written.

Also included: the ZX81 priority table, ZX81 character codes and graphic symbols, systems variables locations, the ZX81 storage plan, digit assignment in flags, the memory map of the ZX81 family, conversion tables for hex-decimal and relative jump count, ZX80 codes, and more.

If you own any one of the Timex/Sinclair micros with 8K ROM—this is the most practical computing helpmate available!

Robert P. Haviland is a professional engineer with extensive experience in the design, construction, and operation of home computers. A well-known author, he has written several books for TAB including *How to Design, Build & Program Your Own Advanced Working Computer System.*

OTHER POPULAR TAB BOOKS OF INTEREST

Using and Programming the ZX81/TS1000, including ready-to-run programs (No. 1617—$7.95 paper; $14.95 hard)

A Kid's Manual for Programming the Sinclair/Timex Computers (No. 1608—$6.95 paper; $12.95 hard)

TAB TAB BOOKS Inc.

Blue Ridge Summit, Pa. 17214

Send for FREE TAB Catalog describing over 750 current titles in print.

FPT > $10.25 ISBN 0-8306-1623-3

PRICES HIGHER IN CANADA 995-1283